

Sistemas baseados em conhecimento

2019.WP.01

white paper

Identificação

Título: White Paper: Sistemas baseados em conhecimento

iD: 2019.WP.01

Projeto: Difusão do Conhecimento

Data: Março de 2019

Local: Brasília, DF

Versão: 1.1

Revisões

Data	Alterações / Comentário	Revisor
2019.02.05	Criação do documento	Mamede Lima-Marques
2019.03.05	versão 1.1	Mamede Lima-Marques

Ficha catalográfica

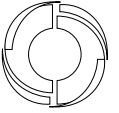
Sistemas baseados em conhecimento / Mamede Lima-Marques. – : Brasília, DF: Instituto Modal, Março de 2019.

30 p. : il. (algumas color.) : 21,0x29,7 cm

White Paper – Instituto Modal de Ciência, Tecnologia e Inovação, Março de 2019. Versão final.

1. Inteligência Artificial 2. Sistemas Especialistas 3. Sistemas Baseados em Conhecimento I. Título

CDD 001.42



Instituto Modal de Ciência, Tecnologia e Inovação

Diretor Presidente

Mamede Lima–Marques

Diretor Técnico

Bruno Carvalho Castro Souza

Diretor Administrativo-financeiro

Wellington de Souza Evangelista

Conselho de Administração

Presidente

José Manuel de Abreu Pita Pombo

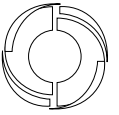
Comitê Científico

Walter Alexandre Carnielli

Mamede Lima–Marques

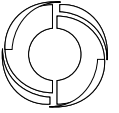
Autores

Mamede Lima–Marques, PhD



Sumário

	Sumário	4
	Introdução	5
1	Sistemas baseados em conhecimento	6
1.1	Sistemas especialistas	7
1.2	Base de conhecimento	8
1.2.1	Linguagens de representação de conhecimento	9
1.2.2	Engenharia de conhecimento	10
1.3	Mecanismo de inferências	12
1.4	Interface homem/máquina	12
1.5	Módulo de explicações	12
1.6	Ferramentas de desenvolvimento	13
2	Exemplo completo: sistema especialista em animais	14
2.1	Base de conhecimento	15
2.2	Mecanismo de inferências	19
2.3	Interface homem/máquina	22
2.4	Módulo de explicações	24
	Considerações finais	28
	REFERÊNCIAS	29
	Sobre o INSTITUTO MODAL	30
	Glossário	30

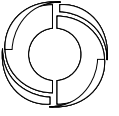


INTRODUÇÃO

Em tempos atuais, toda ação de um projeto é apoiada por conhecimento e experiência; é por isso que os designers tentam definir e coletar o conhecimento de uma forma que possa ser reutilizado em projetos futuros.

Atividades relacionadas à *engenharia de conhecimento* são demoradas, difíceis e muitas vezes caras. Além disso, uma *base de conhecimento* estabelecida pode ser usada depois de algum tempo, o que torna o processo de aquisição de conhecimento raro, e poucas empresas investem na *engenharia de conhecimento*.

O objetivo do estudo apresentado neste trabalho é mostrar como as técnicas vinculadas à *engenharia de conhecimento* e, conseqüentemente, aos *sistemas baseados em conhecimento* podem aumentar a velocidade e a qualidade do processo de desenvolvimento para a classe de problemas que envolve a manipulação e conhecimento.



1 SISTEMAS BASEADOS EM CONHECIMENTO

Um dos principais objetivos de sistemas de Inteligência Artificial (IA) é executar funções que, caso um ser humano fosse executar, seriam consideradas inteligentes. Podemos pensar em algumas características básicas desses sistemas, como a capacidade de raciocínio (aplicar regras lógicas a um conjunto de informações disponíveis para chegar a uma conclusão); aprendizagem (aprender com os erros e acertos de tal maneira que o agir futuro seja mais eficaz; reconhecer padrões (tanto padrões visuais e sensoriais, como também padrões de comportamento); inferência (capacidade de conseguir aplicar o raciocínio em situações do nosso cotidiano); dentre outras. (SBC), se enquadram dentro dessas características.

Portanto, *sistema baseado em conhecimento são 'programas de computador que usam o conhecimento representado explicitamente para resolver problemas. Eles manipulam conhecimento e informação de forma inteligente e são desenvolvidos para serem usados em problemas que requerem uma quantidade considerável de conhecimento humano e de especialização. Tal programa de computador utiliza conhecimento e inferências com o objetivo de resolver problemas de dificuldade que requer uma perícia humana.'*

Em um SBC, conhecimentos são armazenados de maneira tal que o sistema define a forma (sintaxe) e o que significa aquele conhecimento, como ele deve ser interpretado (semântica). Para tanto, fazem uso de *Linguagens de Representação de Conhecimento* (LRC).

SBC combinam qualidades de homens com qualidades de máquinas. Eles tentam “raciocinar” como um homem, criando relações e aprendendo com o passar de suas experiências.

Os SBC estão conquistando espaço entre as tecnologias de informação à medida que o hardware se desenvolve. No ambiente de acirrada competitividade presente nas empresas, esta tecnologia vem apoiá-las estrategicamente na agilização, principalmente, do processo decisório e na solução de diversos problemas considerados complexos.

Apresentamos na [Figura 1](#) uma arquitetura típica de um [sistema baseado em conhecimento](#).

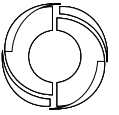
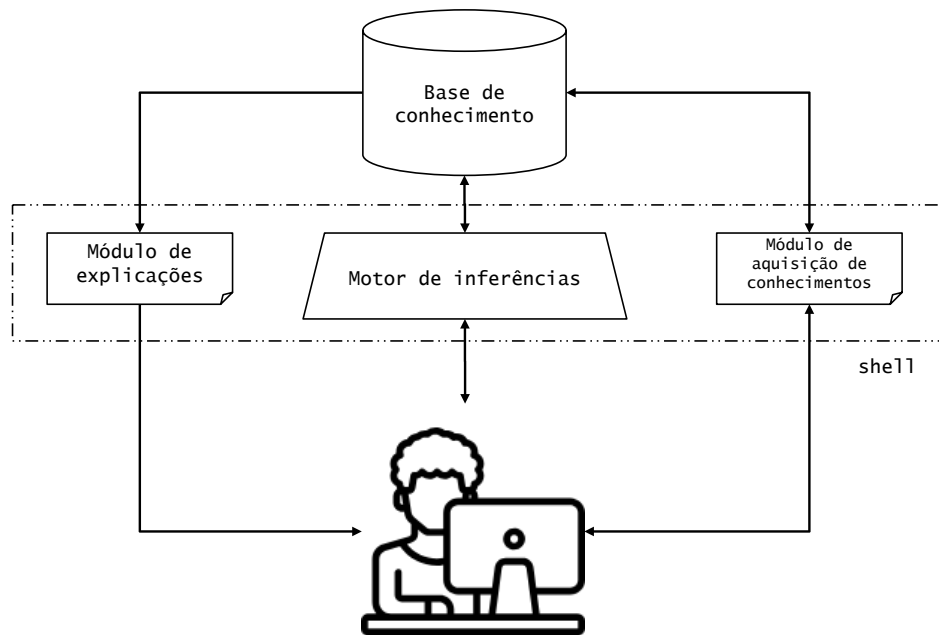


Figura 1 – Arquitetura típica de um sistema baseado em conhecimento



Fonte: o autor

1.1 SISTEMAS ESPECIALISTAS

Quando é construído um sistema baseado em conhecimento voltado para uma área muito específica – absorvendo a experiência de um especialista (ou perito) humano, ele é denominado *Sistema Especialista* encontra-se também a denominação *Sistema Perito*. *Perícia* (do ingl: expertise): a capacidade de aplicar habilidades intelectuais para resolver problemas em domínios específicos, com um desempenho e qualidade de solução superiores à média dos profissionais da área’.

Portanto, sistema especialista é ‘um tipo de sistema baseado em conhecimento, cujo objetivo é simular o raciocínio de um profissional especialista ou perito em alguma área de conhecimento bem específica’. Os conhecimentos necessários para se chegar à solução, bem como os procedimentos de inferências utilizadas podem ser assemelhados a uma modelização da perícia de muitos especialistas no domínio considerado.

Apresentamos na Figura 2 uma arquitetura típica de um sistema especialista.

Sistemas especialistas ‘um tipo de sistema baseado em conhecimento, cujo objetivo é simular o raciocínio de um profissional especialista ou perito em

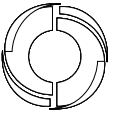
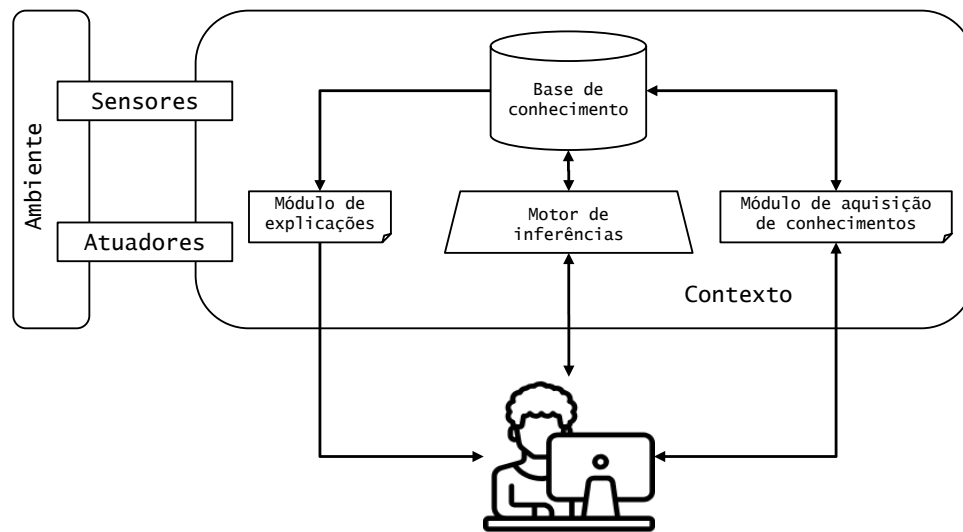


Figura 2 – Arquitetura típica de um sistema especialista



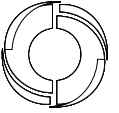
Fonte: o autor

alguma área de conhecimento bem específica'. Por exemplo, um sistema especialista em "câncer de mama" (área específica da medicina) perguntaria certos dados ao usuário e forneceria um diagnóstico acrescido de um aconselhamento profissional sobre o que seria o melhor a fazer nesse caso informado.

Um sistema especialista (*expert system*) é uma classe de programa de computador desenvolvido por pesquisadores de IA no início nos anos 1960 e aplicado comercialmente durante as décadas seguintes. Em síntese, são programas constituídos por uma série de estruturas (com forte tendência a serem *heurísticas*, que analisam informações (normalmente fornecidas pelo usuário do sistema) sobre uma classe específica de problema (ou domínio do problema).

1.2 BASE DE CONHECIMENTO

Um elemento chave em qualquer aplicação de inteligência artificial é o conhecimento. Sobre o *conhecimento* há que se adotar uma corrente filosófica como base de referência epistemológica. Nossa experiência nos leva para à "*fenomenologia*" de Husserl (2001). Neste contexto, o conhecimento só é possível a partir da experiência "sujeito-objeto". Portanto, considera-se conhecimento o conjunto de propriedades do objeto apreendido pelo sujeito. Vamos adotar aqui uma excessão e chamaremos de "conhecimento registrado" o conjunto de especificações contido em uma *base de conhecimento* (ver [subseção 1.2](#) e [subseção 2.1](#)), um dos módulos



característicos de um **sistema baseado em conhecimento**.

O sistema de IA não é baseado em processo algorítmico, mas sim em representação e manipulação simbólica. Usando símbolos é possível criar uma base de conhecimentos que estabelece fatos, conceitos e os relaciona entre eles. O processo é qualitativo e não quantitativo como num algoritmo computacional convencional típico.

A base de conhecimento (do ingl: knowledge base): um repositório de conhecimento registrado utilizando linguagens de representação de conhecimento. A base de conhecimento contém todos os fatos, idéias, relacionamentos e interações de um domínio limitado’.

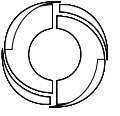
Uma máquina de inferência analisa o conhecimento e deriva conclusões. A interface com o usuário permite que novo conhecimento seja apropriado e implementa a comunicação com o usuário.

Dentre as várias modalidades de classificação de “conhecimento” devemos destacar a **heurística**, que é *‘a arte de descobrir e inventar ou resolver problemas mediante a experiência, somada à criatividade e ao pensamento lateral ou pensamento divergente. Portanto, são procedimentos utilizados quando um problema a ser resolvido é por demais complexo ou traz informações incompletas. No seu sentido como adjetivo, refere-se a coisas concretas, como estratégias heurísticas, regras heurísticas ou silogismos e conclusões heurísticas’.*

Esse tipo conhecimento possui fundamental importância para os **sistema baseado em conhecimento** porque a caracterização de inteligência fica mais evidente quando usa-se expressiva quantidades dos mesmos. Coletar e manipular computacionalmente esse tipo de conhecimento é tarefa da **engenharia de conhecimento**.

1.2.1 LINGUAGENS DE REPRESENTAÇÃO DE CONHECIMENTO

Linguagem de representação de conhecimento (do ingl: knowledge representation language): estruturas de linguagem destinadas à representação do conhecimento, adequadas para serem processadas por computadores. É uma linguagem formal, implementável nas estruturas de dados de um computador digital, destinada a representar todo o conhecimento ou, pelo menos, todo o conhecimento em algum domínio de aplicação de IA. Destina-se como um meio para armazenar conhecimento e para inferência mecanizada em seu domínio. Uma linguagem de representação do conhecimento é o análogo em IA da linguagem do pensamento na ciência cognitiva’.



O ponto de partida para a representação do conhecimento é a *hipótese da representação do conhecimento* formalizada pela primeira vez por [Smith \(1985\)](#):

Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantic attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge.

Existem vários métodos para descrever e registrar o conhecimento com diferentes modos de abordagem. Dentre outros pode-se destacar:

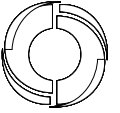
- a. *Redes Semânticas* descrevem relações entre objetos;
- b. *Frames* descrevem o conhecimento em estruturas, em forma de armários (com gavetas), que podem conter em suas gavetas dados, regras, ações, etc.
- c. *Taxonomias* descrevem o conhecimento como um conjunto de conceitos hierarquicamente estruturados.
- d. *Ontologias* descrevem o conhecimento como um conjunto de termos, seus conceitos e as relações entre tais termos.

Para cada situação haverá uma forma mais ou menos adequada para representar o conhecimento. O profissional deverá estruturar cuidadosamente como o conhecimento do sistema a ser representado.

1.2.2 ENGENHARIA DE CONHECIMENTO

As primeiras metodologias da Engenharia de Conhecimento tinham uma abordagem focada na transferência de conhecimento dos indivíduos, basicamente os especialistas que detinham o conhecimento, para formalismos de representação e métodos de raciocínio. Os sistemas construídos buscavam reproduzir o mecanismo de solução de problemas dos especialistas nos sistemas gerados, na mesma forma como as pessoas faziam.

A partir dos anos 90, diversos trabalhos independentes influenciaram os estudos de representação de conhecimento e raciocínio e evoluíram para os novos paradigmas da Engenharia de Conhecimento. Entre esses estudos, os estudos sobre ontologias ([GRUBER, 1992](#); [GRUBER, 1993](#); [GRUBER, 1995](#); [GUARINO, 1995](#);



GUIZZARDI, 2005) e os projetos do programa Europeu Esprit, números P12 e subsequentes, buscavam uma metodologia mais estruturada para o desenvolvimento de sistemas baseados em conhecimento, passível de aplicação comercial.

A base de conhecimento é o repositório do conhecimento do especialista no domínio específico. A base deve ser construída de modo a permitir que o especialista, que não tem conhecimento de programação, tenha acesso ao conhecimento representado, no formato mais próximo possível de sua maneira de expressá-lo.

A engenharia de conhecimento *'busca capturar o conhecimento das organizações e dos indivíduos e formaliza-lo de maneira independente de domínio de modo a permitir sua apropriação e reuso em sistemas ou processos. Busca ainda definir metodologias para cobrir as etapas para apropriação do conhecimento, que são: a) a aquisição do conhecimento de suas fontes humanas ou legadas; b) a modelagem conceitual, ou a associação dos elementos e conceitos cognitivos a construtos de representação; c) a representação dos conhecimentos adquiridos através de um ou mais formalismos capazes de capturar a semântica dos conceitos; e d) sua validação quanto aos objetivos da organização'*.

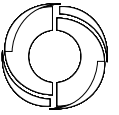
Em décadas passadas, o desenvolvimento da engenharia de conhecimento foi motivado pela incapacidade da indústria de software em reproduzir o sucesso dos seus protótipos ao gerar grandes sistemas especialistas aplicados a problemas reais da indústria. Houve então um esforço com o objetivo de transformar o processo incremental de construir sistemas baseados em conhecimento em uma disciplina da Engenharia baseada em métodos, linguagens e ferramentas especializadas (STUDER; BENJAMINS; FENSEL, 1998).

Capturar essas habilidades, ou perícia, estreitamente relacionadas à aplicação do conhecimento e reproduzi-las através de mecanismos computacionais próprios têm sido o objetivo inicial e maior da engenharia de conhecimento.

É importante ressaltar o postulado de dois dos grandes cientistas em inteligência artificial Lenat e Feigenbaum (1991):

The Knowledge Principle:

"If a program is to perform a complex task well, it must know a great deal about the world in which it operates. In the absence of knowledge, all you have left is search and reasoning, and that isn't enough".



1.3 MECANISMO DE INFERÊNCIAS

O mecanismo de inferências é responsável pelo raciocínio automático usando a *regra de resolução*, como mostra a fórmula (1) abaixo, como expressão fundamental de inferência.

$$\frac{\alpha \vee \beta \quad \delta \vee \neg\beta}{\alpha \vee \delta} \quad (1)$$

Um motor de inferência é um modelo que após ser estimulado com solicitações predeterminadas, oferece as soluções possíveis. Este é o núcleo inteligente de um sistema especialista, onde a capacidade do motor de inferência é baseada numa combinação de raciocínios (inferências). As estratégias usadas pode incluir:

- a) *forma regressiva*: partindo de uma conclusão, feita pelo usuário ou pelo sistema, é feita uma pesquisa por meio do conhecimento acumulado para se provar a afirmação inicial;
- b) *forma progressiva*: respostas fornecidas pelo usuário desencadeando um processo de busca até que se encontre a solução.

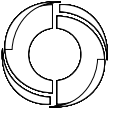
No exemplo apresentado, o mecanismo busca os conhecimentos necessários na base e aplica-os de forma adequada a partir de um encadeamento de busca em profundidade, portanto *forma regressiva*, onde a primeira opção escolhida é a que encontra-se mais à esquerda.

1.4 INTERFACE HOMEM/MÁQUINA

A interface homem/máquina é responsável por toda a interação entre usuário e sistema. Através dela são formatadas e colocadas perguntas para o usuário. As perguntas são formadas a partir de conhecimento advindo da base de conhecimento e de estruturas compatíveis com o português criados neste módulo.

1.5 MÓDULO DE EXPLICAÇÕES

Uma das características dos sistemas especialistas é que eles devem explicar seu próprio raciocínio em tempo de execução. Como normalmente o raciocínio



envolvido é bem complexo e de difícil entendimento, os módulos de explicação cumprem um importante papel na tarefa de esclarecer e acompanhar as etapas da argumentação envolvida. Neste sistema, o módulo de explicação armazenada todas as conclusões intermediárias e, utilizando-se do conhecimento da base, prepara-os para formar explicações que o usuário possa entender.

1.6 FERRAMENTAS DE DESENVOLVIMENTO

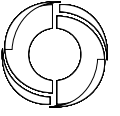
Historicamente foram desenvolvidas diversas ferramentas com o intuito de auxiliar no processo de implementação de **sistema especialista**. Dentre elas pode-se destacar a linguagem de programação **Prolog**: *'(programming in logic) trata-se de uma linguagem de programação que se enquadra no paradigma de Programação em Lógica Matemática. É uma linguagem de propósito geral que é especialmente associada com a inteligência artificial e linguística computacional'*

Prolog tem suas raízes na Lógica de Primeira Ordem, uma lógica formal e, ao contrário de muitas outras linguagens de programação, **Prolog** é principalmente uma linguagem de programação declarativa: a lógica do programa é expressa em termos de relações, representadas como fatos e regras. Uma computação é iniciada executando uma consulta sobre essas relações.

A linguagem foi inicialmente concebida por Alain Colmerauer e seu grupo em Marselha, na França, no início dos anos 1970, e o primeiro sistema **Prolog** foi desenvolvido em 1972 por Colmerauer com Philippe Roussel.

Prolog é completo. Isso significa que ele pode fazer exatamente as mesmas coisas que todas as outras linguagens completas podem fazer (o que inclui quase todas as linguagens de programação de todos os tempos), e todas essas outras linguagens podem fazer o que o **Prolog** faz. A única diferença está em quão fácil (ou produtivo) é descrever um algoritmo.

Prolog foi uma das primeiras linguagens de programação lógica, e continua sendo a mais popular entre as linguagens atuais, com várias implementações gratuitas e comerciais disponíveis. A linguagem tem sido usada para prova de teoremas, sistemas especialistas, reescrita de termos, sistemas de tipos, e planejamento automatizado, bem como seu campo de uso original pretendido, processamento de linguagem natural, dentre outras. Os ambientes atuais do **Prolog** suportam a criação de interfaces gráficas com o usuário, bem como aplicativos administrativos e em rede.



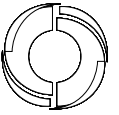
2 EXEMPLO COMPLETO: SISTEMA ESPECIALISTA EM ANIMAIS

Inspirado na implementação da proposta de [Winston \(1984, pp. 119-126\)](#) um sistema especialista em animais foi desenvolvido a partir de uma crítica ao modelo de [SAINT-DIZIER, 1990, pp. 140-147](#)). Foi criada uma nova abordagem e realizada a construção de um novo modelo apoiado em uma completa reengenharia na proposta original de [Saint-Dizier \(1990\)](#). O objetivo foi produzir um sistema que tivesse todas as características de um sistema especialista profissional. Uma primeira versão dessa proposta apareceu em [\(LIMA-MARQUES, 1990\)](#).

Trata-se de um sistema baseado em regras (também conhecido como *rules based expert system*) que, neste caso específico, procura reconhecer um animal a partir da identificação de algumas de suas características. Tais características são solicitadas ao usuário que responde a um questionário até que o sistema alcance sua conclusão.

Esse [sistema especialista](#) foi construído com o objetivo de atender à um curso de nível de graduação, obedecendo a princípios didáticos bem definidos, tais como: simplicidade, objetividade, aderência aos fundamentos da área, base de conhecimento muito reduzida e completude (possui todos os módulos de um sistema profissional).

Cada um dos módulos que aparecem na [Figura 2](#) será desenhado, explicado e sua implementação apresentada nas subseções a seguir.



2.1 BASE DE CONHECIMENTO

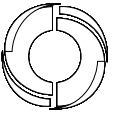
A base de conhecimento do sistema especialista em animais foi desenvolvida buscando usar um formalismo próximo da linguagem natural, com objetivo de facilitar o trabalho de aquisição de conhecimento.

Nesta proposta, a base de usa **regras de produções** como linguagem de representação de conhecimento, em um formato muito próximo da linguagem natural. Para tanto, é utilizada uma estrutura baseada em operadores, que permitem a representação em um formato mais adequado.

*Regras de produções, (do ingl: production rule): um tipo de linguagem de representação de conhecimento, formada por estruturas condição-ação, do tipo **se** condição (ou premissa ou antecedente) ocorre **então** ação (resultado, conclusão ou consequente) deverá ocorrer. Em geral, uma regra de produção pode ser composta por múltiplos antecedentes ligados pelos conectivos lógicos E e OU (ou ambos). O consequente de uma regra de produção também pode ter múltiplas ações'.*

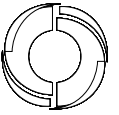
Código 1 – Base de conhecimento

```
1 %-----
2 % Contem : Base de conhecimento
3 %-----
4
5 %-----
6 % Preparacao da Sintaxe das Regras de Producao
7 :- op(900, xfy, ::).
8 :- op(50, fx, regra).
9 :- op(350, fy, se).
10 :- op(300, xfy, e).
11 :- op(500, xfy, entao).
12 %-----
13 % Regras de Producao
14
15 regra 1 ::
16 se      'possui pelos'           e
17         'e'' de regime carnivoro' e
18         'alimenta-se de peixes' e
19         'possui a cor cinza'
20 entao  foca.
21
```



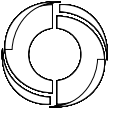
Código 1 (Cont.): Base de conhecimento

```
22 regra 2 ::
23 se      'possui pelos'           e
24         'e'' de regime carnívoro' e
25         'vive na África'         e
26         'alimenta-se de plancton'
27 entao  baleia.
28
29 regra 3 ::
30 se      'possui pelos'           e
31         'e'' de regime carnívoro' e
32         'vive na África'         e
33         'possui juba'
34 entao  leão.
35
36 regra 4 ::
37 se      'possui pelos'           e
38         'e'' de regime carnívoro' e
39         'vive na África'         e
40         'possui a pele manchada' e
41         'sobe em árvores'
42 entao  leopardo.
43
44 regra 5 ::
45 se      'possui pelos'           e
46         'e'' de regime carnívoro' e
47         'vive na floresta'       e
48         'alimenta-se de peixes'  e
49         'e'' branco em alguma parte' e
50         'e'' castanho em alguma parte' e
51         'sobe em árvores'
52
53 entao  urso.
54
55 regra 6 ::
56 se      'possui pelos'           e
57         'vive na África'         e
58         'alimenta-se de plantas'  e
59         'possui a pele manchada'
```

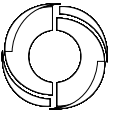
Código 1 (Cont.): Base de conhecimento

```
60
61  entao girafa.
62
63  regra 7 ::
64  se      'possui pelos'           e
65         'vive na Africa'         e
66         'possui prezas de marfim' e
67         'possui trompa'          e
68         'possui a cor cinza'
69
70  entao elefante.
71
72  regra 8 ::
73  se      'possui penas'           e
74         'voa'                     e
75         'possui patas longas'     e
76         'se alimenta de peixes'  e
77         'e'' branco em alguma parte' e
78         'possui a cor cinza'
79  entao garca.
80
81  regra 9 ::
82  se      'possui penas'           e
83         'voa'                     e
84         'se alimenta de peixes'  e
85         'ninha em rochedos'      e
86         'e'' branco em alguma parte' e
87         'possui a cor cinza'
88  entao gaivota.
89
90  regra 10 ::
91  se      'possui penas'           e
92         'possui longo pescoco'    e
93         'possui plumas volumosas' e
94         'possui patas longas'
95  entao avestruz.
96
97  regra 11 ::
```



Código 1 (Cont.): Base de conhecimento

```
98 se      'possui penas'      e
99      'e'' aquatico'      e
100      'se alimenta de peixes'      e
101      'e'' preto e branco'
102 entao   pinguim.
103 %------
```

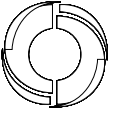


2.2 MECANISMO DE INFERÊNCIAS

O mecanismo de inferências do sistema especialista em animais foi desenvolvido segundo parâmetros de independência de contexto. Isto significa que o mecanismo de inferências juntamente com interface homem-máquina e o módulo de explicações funcionam como um *shell*, ou seja, basta usar a mesma estrutura da base de conhecimento, mas com outro conteúdo, que o modelo deverá funcionar adequadamente.

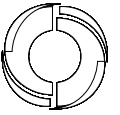
Código 2 – Mecanismo de inferências

```
1 %-----
2 % Contem : Motor de Inferencia
3 %-----
4 :- dynamic descoberto/2.
5 %-----
6 % Identificacao do animal
7 % -----
8
9 identificar :-
10     write('Este eh um Sistema Especialista em animais'),
11     nl,
12     write('O objetivo eh descobrir o nome do animal'),
13     nl,
14     write(' que voce esta observando.'),
15     nl,nl,nl,
16     regra Num :: se Conds entao Conc,
17     colocar_questao(Num, Conds),
18     prova_animal(Conds),
19     explicar(Num, Conds, Conc),
20     apaga_tudo(descoberto/2).
21
22 identificar:-
23     bagof(X, descoberto(X, sim), Lsim),
24     bagof(X, descoberto(X, nao), Lnao),
25     nl,
26     nl,
27     write('Foi demonstrado que o animal:'),
28     nl,
29     mostra_lista(Lsim),
```



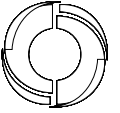
Código 2 (Cont.): Mecanismo de inferências

```
30     write('E ainda, que o animal NAO possui'),
31     nl,
32     write('as seguintes características:'),
33     nl,
34     mostra_lista_nao(Lnao),
35     write('-----'),
36     nl,
37     write('Conclusao: Animal desconhecido pelo sistema.')
38     ,
39     nl,
40     write('-----'),
41     nl,
42     apaga_tudo(descoberto/2).
43
44 %-----
45 % Prova a existencia real do animal segundo as
46 % regras estabelecidas
47 %-----
48 prova_animal(Conds) :-
49     bagof(X, descoberto(X, sim), LConds),
50     compara_cond(Conds, LConds).
51
52 compara_cond(Cond, LConds) :-
53     atom(Cond),
54     !,
55     membro(Cond, LConds).
56
57 compara_cond((Cond e Conds), LConds) :-
58     membro(Cond, LConds),
59     compara_cond(Conds, LConds).
60
61 %-----
62 % perguntas ao usuario
63 %-----
64 colocar_questao(Num, Cond) :-
65     atom(Cond),
66     !,
```



Código 2 (Cont.): Mecanismo de inferências

```
67     questionar(Num, Cond).
68
69 colocar_questao(Num, (Cond e Conds)):-
70     questionar(Num, Cond),
71     colocar_questao(Num, Conds).
72
73 questionar(_, Cond):-
74     user:descoberto(Cond, sim),
75     !.
76
77 questionar(_, Cond):-
78     user:descoberto(Cond, nao),
79     !,
80     fail.
81
82 questionar(Num, Cond):-
83     questao(Num, Cond).
84
85 %-----
86 % Basicos
87 %-----
88 membro(X, [X|_]).
89 membro(X, [_|T]) :-
90     membro(X,T).
91
92 %-----
93 % Apaga toda a base de nome=Funtor e Aridade
94 %-----
95 apaga_tudo(Funtor/Aridade) :-
96     functor(Termo, Funtor, Aridade),
97     retract(Termo),
98     fail.
99 apaga_tudo(_).
100 %-----
```



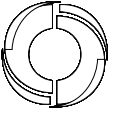
2.3 INTERFACE HOMEM/MÁQUINA

A interface homem/máquina do sistema especialista em animais foi idealizada para capturar questões a partir da base de conhecimento, buscando reconhecer a estrutura de regras de produção e transformá-la em questões que são colocadas para o usuário solicitando respostas.

A interface homem/máquina interage com a base de conhecimento, o mecanismo de inferências e o módulo de explicações.

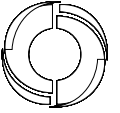
Código 3 – mecanismo de inferências

```
1  %-----  
2  % Contem : Interface Homem/Maquina  
3  %-----  
4  questao(Num, Cond) :-  
5      write('Este animal '),  
6      write(Cond),  
7      write('?'),  
8      nl,  
9  resposta(Num, Cond).  
10 %-----  
11 % Analise da Resposta  
12 %-----  
13 resposta(Num, Cond) :-  
14     read(Resp),  
15     tratar_resp(Num, Cond, Resp).  
16 tratar_resp(Num, Cond, nao) :-  
17     assert(descoberto(Cond, nao)),  
18     !,  
19     fail.  
20 tratar_resp(Num, Cond, sim) :-  
21     assert(descoberto(Cond, sim)),  
22     !.  
23 tratar_resp(Num, Cond, por_que) :-  
24     !,  
25     expl_por_que(Num),  
26     questao(Num, Cond).  
27 tratar_resp(Num, Cond, _) :-  
28     write('-----'),  
29     nl,
```



Código 3 (Cont.): mecanismo de inferências

```
30     write('Resposta incorreta!'),  
31     nl,  
32     write('Por favor responda (sim, nao ou por_que)'),  
33     nl,  
34     write('-----'),  
35     nl,nl,  
36     questao(Num, Cond).  
37 %-----
```

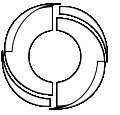


2.4 MÓDULO DE EXPLICAÇÕES

O módulo de explicações do sistema especialista em animais é dinâmico e funciona em tempo de execução, ou seja, pode fornecer explicações no momento em que o sistema está executando. Ainda, oferece explicações sobre a conclusão do seu raciocínio ao final do processo de raciocínio.

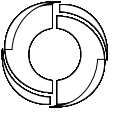
Código 4 – Módulo de explicações

```
1 %-----
2 % Contem : Modulo de Explicacoes
3 %-----
4
5 % explicar(+Num, +Cond, +Conc)
6 explicar(_, Cond, Conc) :-
7     write('-----'),
8     nl,
9     write('Se trata do animal: '),
10    write(Conc),
11    nl, nl,
12    write('Isto foi deduzido a partir de: '),
13    nl, nl,
14    write('O animal: '),
15    nl,
16    explicacao(Cond),
17    write('-----'),
18    nl.
19 % explicacao(+Cond)
20 explicacao(Cond) :-
21     atom(Cond),
22     tab(10),
23     write(Cond),
24     nl,
25     !.
26 explicacao(Cond1 e Cond2) :-
27     tab(10),
28     write(Cond1),
29     nl,
30     explicacao(Cond2).
31 %-----
```

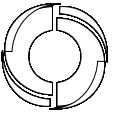
Código 4 (Cont.): Módulo de explicações

```
32 % Explicacoes para o "por_que"
33 % -----
34
35 % expl_por_que(+Num)
36 expl_por_que(Num) :-
37     write('-----'),
38     nl,
39     ja_demonstrei,
40     regra Num :: se Conds entao Conc,
41     write('Agora estou tentando demonstrar a regra '),
42     write(Num),
43     nl,
44     tab(4),
45     write('se o animal tambem:'),
46     nl,
47     monta_regra(Conds),
48     nl,
49     tab(4),
50     write('entao se trata do animal '),
51     write(Conc),
52     nl,
53     write('-----'),
54     nl,nl.
55 %-----
56 % Escreve as condicoes ja demonstradas
57 % -----
58
59 ja_demonstrei :-
60     \+ descoberto(_,_),
61     !,
62     write('Eu ainda nao demonstrei nenhuma regra.'),
63     nl,
64     write('Por favor nao brinque comigo!'),
65     nl,nl.
66
67 ja_demonstrei :-
68     write('Eu ja'' demonstrei que o animal:'),
69     nl,
```



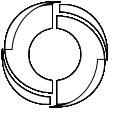
Código 4 (Cont.): Módulo de explicações

```
70     descoberto(Cond, sim),
71     tab(12),
72     write(Cond),
73     nl,
74     fail.
75
76 ja_demonstrei :-
77     nl.
78 %-----
79 % Escreve uma regra na tela
80 % monta_regra(+Cond)
81
82 monta_regra(Cond) :-
83     atom(Cond),
84     tab(12),
85     write(Cond),
86     nl,
87     !.
88
89 monta_regra(Cond1 e Cond2) :-
90     escreve_conds(Cond1),
91     monta_regra(Cond2).
92
93 escreve_conds(Cond1) :-
94     descoberto(Cond1, sim),
95     !.
96
97 escreve_conds(Cond1) :-
98     tab(12),
99     write(Cond1),
100    write(' e'),
101    nl.
102 %-----
103 % Escreve uma lista na Tela
104 % mostra_lista(+Lista)
105
106 mostra_lista([]).
107
```



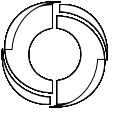
Código 4 (Cont.): Módulo de explicações

```
108 mostra_lista([Cond|Conds]) :-  
109     tab(12),  
110     write(Cond),  
111     nl,  
112     mostra_lista(Conds).  
113  
114 mostra_lista_nao([]).  
115  
116 mostra_lista_nao([Cond|Conds]) :-  
117     tab(12),  
118     write('nao '),  
119     write(Cond),  
120     nl,  
121     mostra_lista_nao(Conds).  
122 %-----
```



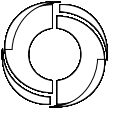
CONSIDERAÇÕES FINAIS

O objetivo desse *white paper* foi resumir de forma prática algumas das principais características dos **sistemas baseados em conhecimento**. O modelo de **sistemas especialistas** em animais seguiu uma implementação particular desenvolvida pelo autor. A intenção é contribuir para que o processo de tomada de decisão para o seu uso em casos específicos seja relevante.



Referências

- GRUBER, T. R. *Ontolingua: A mechanism to support portable ontologies*. [S.l.]: Stanford University, Knowledge Systems Laboratory Stanford, 1992. Citado 2 vezes nas páginas 10 e 11.
- GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge acquisition*, Elsevier, v. 5, n. 2, p. 199–220, 1993. Citado 2 vezes nas páginas 10 e 11.
- GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, Elsevier, v. 43, n. 5-6, p. 907–928, 1995. Citado 2 vezes nas páginas 10 e 11.
- GUARINO, N. Formal ontology, conceptual analysis and knowledge representation. *International journal of human-computer studies*, Elsevier, v. 43, n. 5-6, p. 625–640, 1995. Citado 2 vezes nas páginas 10 e 11.
- GUIZZARDI, G. *Ontological foundations for structural conceptual models*. Enschede, The Netherlands: Universal Press, 2005. (CTIT PhD Thesis Series). Citado 2 vezes nas páginas 10 e 11.
- HUSSERL, E. *Logical Investigations*. London: Routledge, 2001. (International Library of Philosophy, v. 1). ISBN 9780415241892. Citado na página 8.
- LENAT, D. B.; FEIGENBAUM, E. A. On the thresholds of knowledge. *Artificial Intelligence*, Elsevier, v. 47, n. 1-3, p. 185–250, 1991. Citado na página 11.
- LIMA-MARQUES, M. Introduction a l'Intelligence artificielle et au langage PROLOG. Toulouse, França, Apostila. 1990. Citado na página 14.
- SAINT-DIZIER, P. *An Introduction to Programming in Prolog*. New York: Springer-Verlag, 1990. Translated by Sharon J. Hamilton. Citado na página 14.
- SMITH, B. C. Prologue to reflections and semantics in a procedural language. In: BRACHMAN, R.; LEVESQUE, H. J. (Ed.). *Readings in knowledge representation*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1985. p. 31–40. ISBN 978-0-934613-01-9. Citado na página 10.
- STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge engineering: Principles and methods. *IEEE Transactions on Data and Knowledge Engineering*, v. 25, n. 1-2, p. 161–197, 1998. Citado na página 11.
- WINSTON, P. H. *Artificial Intelligence*. 2. ed. London: Addison-Wesley, 1984. Citado na página 14.



SOBRE O INSTITUTO MODAL

O INSTITUTO MODAL é uma *Instituição de Ciência e Tecnologia* (ICT), de natureza privada e sem fins lucrativos, que surgiu da convergência entre pesquisadores da área de informação e de tecnologia, empresários e profissionais com larga experiência no setor privado. Essa junção permitiu construir pontes entre fundamentação teórica e soluções reais, viabilizando a o uso da experiência científica às necessidades do mercado e da sociedade e encontrando soluções para problemas dos mais diversos tipos.

O INSTITUTO MODAL tem por objeto a realização de pesquisa básica e aplicada de caráter científico ou tecnológico e o desenvolvimento de novos produtos, serviços ou processos voltados prioritariamente ao objeto “informação”, zelando pelo reconhecimento da importância da inovação no sistema produtivo nacional.

Para saber mais, visite institutomodal.org.br.

Contatos podem ser feitos pelo e-mail modal@modal.org.br.